REAL-TIME PAYLOAD CONTROL AND MONITORING ON THE WORLD WIDE WEB

Charles Sun [†] *and May Windrem* [††]

NASA Ames Research Center
MS 244-19
Moffett Field, CA 94087, USA
FAX: 415-604-0673, [†] *E-mail: csun@mail.arc.nasa.gov*
[††] *E-mail: mwindrem @mail.arc.nasa.gov*

ABSTRACT

World Wide Web (W3) technologies such as the Hypertext Transfer Protocol (HTTP) and the Java object-oriented programming environment offer a powerful, yet relatively inexpensive, framework for distributed application software development. This paper describes the design of a real-time payload control and monitoring system that was developed with W3 technologies at NASA Ames Research Center. Based on Java Development Toolkit (JDK) 1.1, the system uses an event-driven "publish and subscribe" approach to inter-process communication and graphical user-interface construction. A C Language Integrated Production System (CLIPS) compatible inference engine provides the back-end intelligent data processing capability, while Oracle Relational Database Management System (RDBMS) provides the data management function. Preliminary evaluation shows acceptable performance for some classes of payloads, with Java's portability and multimedia support identified as the most significant benefit.

1. INTRODUCTION

The Space Station Biological Research Project (SSBRP) at NASA Ames Research Center is responsible for Life Sciences research onboard the International Space Station. SSBRP payloads include host systems and habitats. Examples of host systems are Centrifuge, Glovebox, and Habitat Housing Rack. Examples of habitats are Insect Habitat, Aquatic Habitat, and Cell Culture Unit. Because the research experiments involve many Principal Investigators (PIs) worldwide, providing cost-effective real-time data feed and data visualization capabilities to these PIs pose a significant engineering challenge. One solution is to leverage existing World Wide Web (W3) technologies such as Hypertext Transfer Protocol (HTTP) and the Java object-oriented programming environment. These technologies are universal (open standard) and low cost. This paper describes the design and implementation of a SSBRP payload control and monitoring system prototype that leverages these technologies, along with the lessons learned.

2. THE OPERATION CONCEPT

Figure 1 shows the proposed system operation. The telemetry downlink provides the data as UDP/IP packets to the User Operations Facility (UOF) at Ames Research Center. The maximum data rate is expected to be 10Mbps, including compressed digital video. A subset of the data is repackaged as TCP/IP packets at the UOF and forwarded to PIs in remote locations via the Internet. For command up-link, PIs request commands at their respective remote locations; the UOF then authorizes those requests.

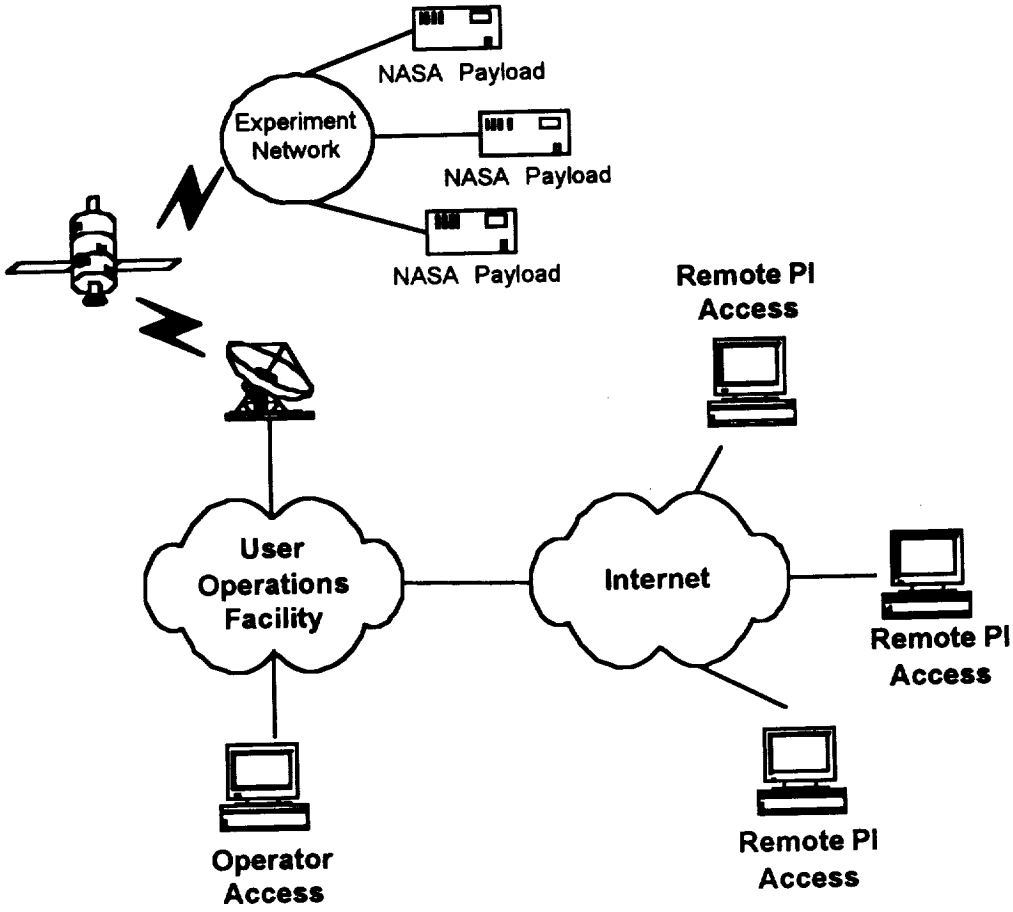

Figure 1. The SSBRP Operation Concept

3. SOFTWARE REQUIREMENTS

The nature of SSBRP operations suggests the following software requirements for the UOF:

- Graphical User Interface
- Snapshot & Motion Video (MPEG) Support
- Rule-based Intelligence for Automatic Problem Detection
- Data Management using Industry-Strength RDMS
- Platform Independent Distributed Clients
- Internet TCP/IP Data Distribution
- On-Orbit Configuration Management Support

4. A REVIEW OF EXISTING SOLUTIONS

In the summer of 1997, we evaluated several Commercial Off The Shelf (COTS) software packages using the software requirements as evaluation criteria. Table 1 summarizes our findings. In conclusion, we found that the Internet W3-based Java programming environment to be better suited for our requirements than X-Windows based (RTworks, G2, Sammi) or Microsoft Windows based (LabWindows, LabView) COTS frameworks.

	GUI	Video	RuleBase	Database	Platform	Network	Config
RTworks	Good	N/A	Good	Fair	Good	Good	N/A
G2	Fair	N/A	Good	N/A	N/A	Fair	N/A
Sammi	Good	N/A	N/A	N/A	N/A	Fair	N/A
LabView	Good	N/A	N/A	N/A	Fair	N/A	N/A
LabWindows	Fair	N/A	N/A	Fair	N/A	Fair	N/A
Java	Good	Fair	Fair	Good	Good	Good	Fair

Table 1. A Comparison of Existing Software Frameworks with Java
(Note: N/A means Not Available.)

5. SYSTEM DESIGN

5. 1 THE REAL-TIME CONTROL AND MONITORING SUBSYSTEM

This subsystem is responsible for distributing data and generating commands in real-time for UOF operators and PIs at remote locations. An event-driven "publish-and-subscribe" model is used to create a highly scalable, loosely coupled framework that can be easily modified for different science experiments over the lifetime of the Space Station. This framework is depicted in Figure 2 using Unified Modeling Language (UML) notation.

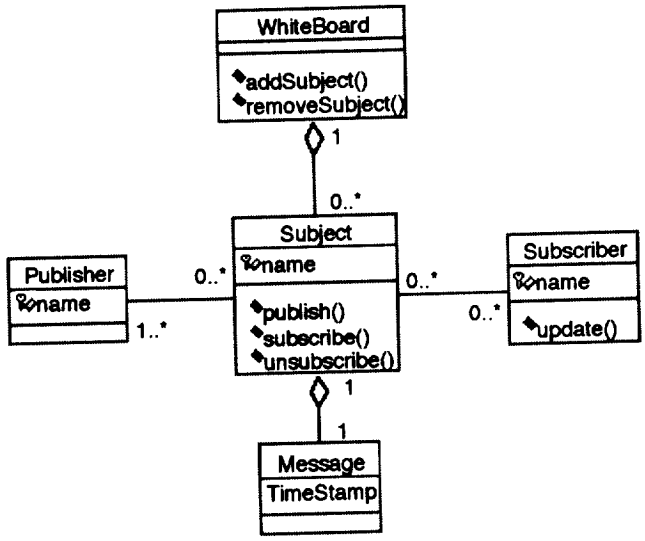

Figure 2. A "White Board" Publish-and-Subscribe Object Model

Under this framework, there are three major types of entities (classes):

1. White Board (WB), the event-passing server.
2. Data Publisher (DP), the data or command event source.
3. Data Subscriber (DS), the event recipient.

The White Board (WB) is the centerpiece of the model. A WB contains many Subjects defined by DPs. The messages passed from a DP to DSs are associated with unique Subject names. A message can be Simple or Composite. A Simple Message contains one object such as Double, String, Integer. A Composite Message contains a serialized collection of objects.

The following sequence of events takes place in a typical message passing operation between a DP and a DS:

1. DP defines a Subject S on WB (this causes a new Subject to be created).
2. DS subscribes to Subject S on WB.
3. DP publishes a message related to Subject S.
4. Subject S forwards the message to all DS who subscribe to Subject S.

This interaction is shown in the UML Object Interaction diagram in Figure 3 below.

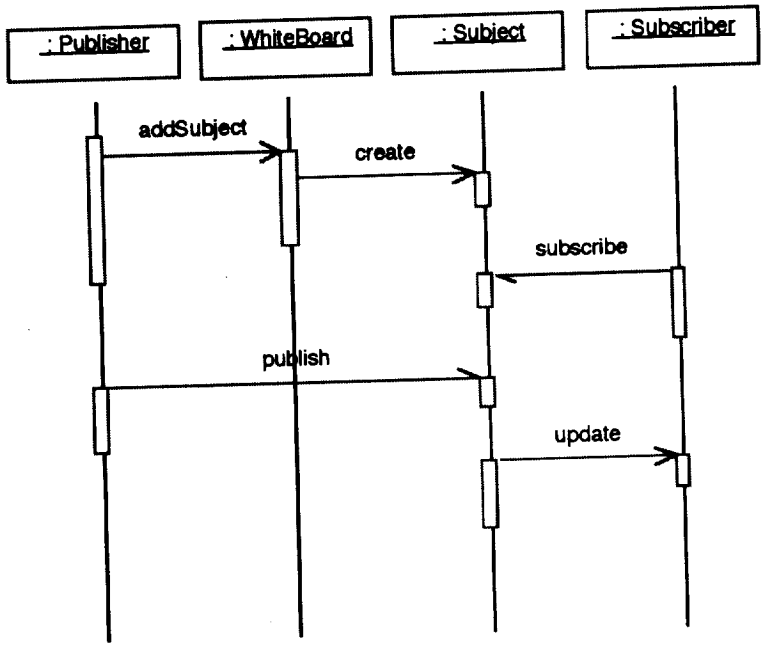

Figure 3. A "White Board" Publish-and-Subscribe Event Model

5.2 THE DATA STORAGE AND RETRIEVAL SUBSYSTEM

This subsystem is responsible for storing real-time data and distributing stored data via the Internet. Because the subsystem's operation is very similar to transaction-oriented business information systems, a standard client/server distributed database framework is used. The subsystem stores video data as files, which are indexed by time-stamped entries in the database for retrieval.

6. THE JAVA SOLUTION

The first Java prototype was completed using JDK 1.1 in December of 1997. The development effort took two software engineers less than 4 months, because the Java programming environment provides many timesaving component packages. Furthermore, the availability of Integrated Development Environment (IDE), such as Java Workshop and Borland Jbuilder, simplified many Graphical User Interface construction tasks and significantly reduced our development time. The utilization of Java JDK and 3rd party packages is summarized in Table 2.

Problem	Solution
White Board Framework	RMI
GUI Construction	JavaBean
Configuration Management	Object Serialization
Motion Video	Java Media Framework
Intelligent Agent	JESS
Database integration	JDBC

Table 2. Selected Java JDK 1.1 and 3rd Party Packages

6.1 THE WHITE BOARD FRAMEWORK BASED ON RMI

Remote Method Invocation (RMI) is a set of classes that enables Java programs to communicate with each other on the object level across a TCP/IP network. RMI hides the details of byte-level "socket" communication from Java programmers. The "White Board" abstract model for Inter-Process Communication (IPC) was constructed using Java RMI and this turned out to be a simple task because the RMI abstraction was flexible and easy-to-use.

6.2 GUI CONSTRUCTION USING JAVABEANS

JavaBeans is a standard for re-usable Java components. Most Java IDE tools accept the standard and provide programmers visual "drag and drop" capabilities using JavaBeans. A set of JavaBeans components were developed specifically for control and monitoring tasks and then assembled using Borland JBulder software tool. Figure 4 and Figure 5 show sample GUI Applets constructed using the JavaBeans technique.

6.3 CONFIGURATION MANAGEMENT USING OBJECT SERIALIZATION

Object Serialization is a convenient way to package many objects together into one large object for transporting through RMI or persistent storage. One important task for the UOF is tracking the location of each habitat and the overall configuration of the onboard hosts. To simplify the task of updating and storing configuration history, the configuration of the entire payload facility is represented as a tree data-structure and combined together as a single object using the Object Serialization feature of JDK 1.1.

6.4 MOTION VIDEO INTEGRATION USING JAVA MEDIA FRAMEWORK

The system's video playback capability is achieved using Java Media Framework (JMF) which supports the MPEG-1 format and HTTP data transfer. Currently, JMF 1.0 is available only on three computer platforms (Sun, Intel, and SGI). MPEG-2 support in future JMF release is unknown.

6.5 INTELLIGENT AGENT USING JESS

The rule-based intelligent agent under development is based on JESS (Java Expert System Shell), a Java port of CLIPS. Because many CLIPS documents and sample codes are available in the public domain, JESS was our choice for writing the agent under Java. The agent works as a watchdog and data limit detector in the background. Should a problem be detected, the agent would generate a warning message. In terms of the White Board framework, the agent is configured as a subscriber for certain raw data sets and as a publisher for warning messages.

6.6 DATABASE INTEGRATION USING JDBC

Java Database Connectivity (JDBC) is the standard interface for Java programs to connect to a Relational Database. There are two types of JDBC implementations: 2-tier JDBC Bridge, and 3-tier networked JDBC. A JDBC bridge connects to a database server on the local host while a networked JDBC connects to a database server on a remote host. The prototype uses a networked JDBC implementation from WebLogic Inc. for connecting to Oracle 7 running on a SGI server. The database supports real-time data storage and off-line data retrieval.

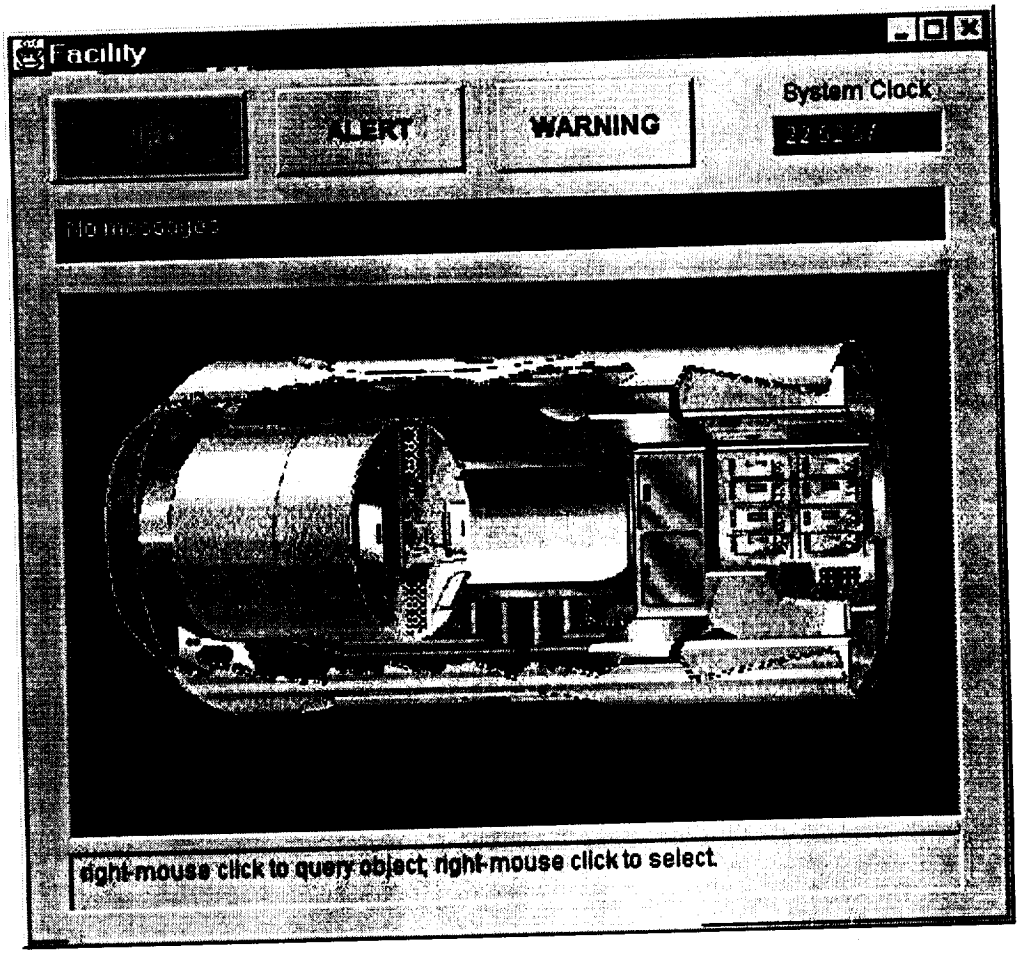

Figure 4. A Sample User-Interface Display (Top Level)

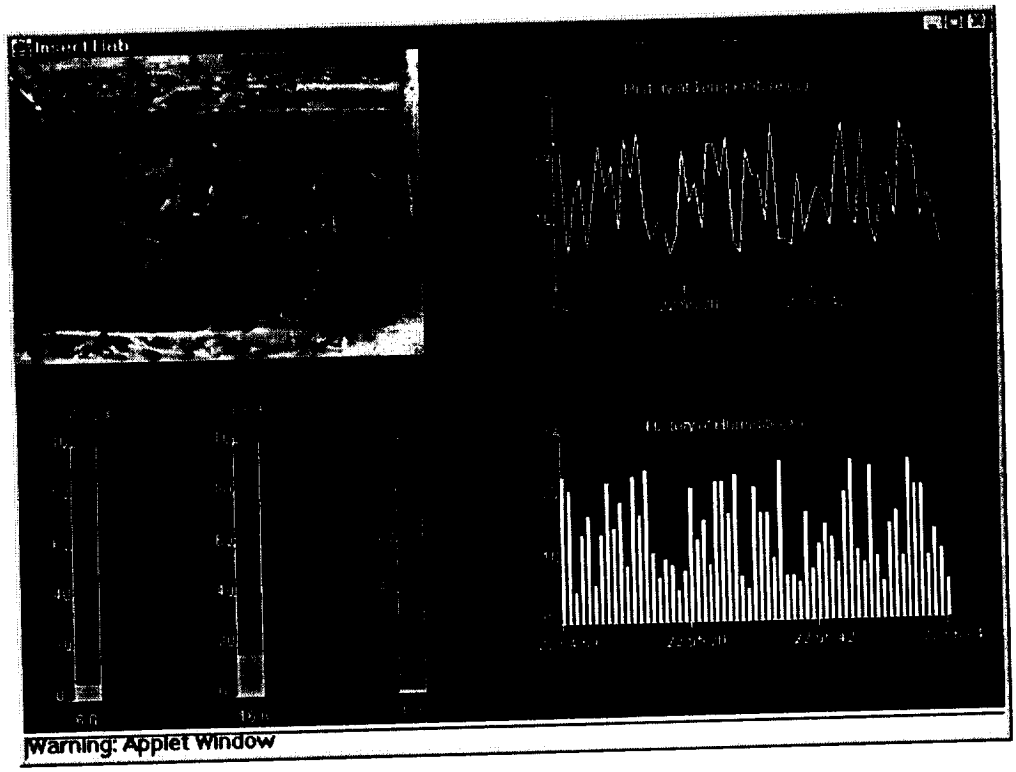

Figure 5. A Sample User-Interface Display (Detailed Level)

7. LESSONS LEARNED

- Acceptable Performance for Low-Bandwidth Applications

Our test showed that Java provides acceptable performance for low-bandwidth applications, even under the interpreted environment of Java Virtual Machine. Under the constant load of a few Kbytes per second, our client-side application running on a Pentium/166 notebook PC on a 10BaseT LAN encountered no performance problems. This result encourages us to adapt Java to support the upcoming SSBRP mission, which has a small set of data points and a low sampling rate.

- Low-Cost Platform-Independent Software Development

We were pleasantly surprised by the high quality of free Java JDK software provided by Sun Microsystems and Silicon Graphics. Very few portability problems were encountered when we executed the client-side Applet using the "appletviewer" program on different computer platforms.

- Applet Deployment Difficulties within Web Browser Software

Although our Java "Applet" performs consistently within the "appletviewer" program, it encounters problems running inside some Web Browser software. For example, Microsoft Internet Explorer does not support RMI because RMI competes with Microsoft DCOM standard. Apparently, this is a well-known problem and at least three solutions exist today: (1) launch Applet with "appletviewer" (2) install Java Activator, and (3) run the client program as an application with Zero Administration Framework.

- Security Model Incompatibilities

Object Signing is problematic because many different standards exist. JDK Object Signing will not work within Netscape Communicator. Netscape Object Signing will not work with Microsoft Internet Explorer. We will wait until further in the design process to allow the differences in the standards to be resolved or for a single standard to be accepted.

- Non-Platform-Independent Features

Many of JDK's most exciting new features, such as video playback and speech recognition, are not truly platform-independent at this time. For example, Java Media Framework (JMF) is limited to the Intel, Sun, and SGI platforms. Speech Framework is limited to the Intel platform. Multimedia features are thus more useful in the context of Local Area Network deployment, in which the client platform and bandwidth availability are well defined.

8. FUTURE DIRECTION

- Improving Event-Based IPC Model

We plan to incorporate sophisticated capabilities such as Access Control, Load Balancing and Fault Tolerance into our event-based IPC model. After we completed the first prototype in December of 1997, two commercial Publish-and-Subscribe IPC frameworks appeared on the market: Tengah Even Model and JavaSpace. These frameworks provide a higher level of IPC abstraction than RMI, and thus offer potential savings in development time.

- Improving Client-side Software Deployment Environment

Because of the incompatibility problems with various Web browser software, we are searching for an alternative to the existing "Applet" environment for client-side software deployment. The two candidates are Java Activator, and Zero Administration Client (ZAC). Java Activator allows the latest version of JDK to be installed as a plug-in module independent of browser software. ZAC provides a zero-maintenance framework for running Java application on the client-side, and completely removes the deployment dependency on browser software.

REFERENCES

1. Space Station Biological Research Project (SSBRP) http://pyrocis.arc.nasa.gov
2. HyperText Transfer Protocol. http://www.w3.org/hypertext/WWW/Protocols
3. Java Programming Language. http://www.javasoft.com
4. RTworks software framework. Talarian Corp. http://www.talarian.com
5. G2, Gensym Corporation. http://www.gensym.com
6. Sammi, Kinesix Corporation. http://www.kinesix.com
7. LabView & LabWindows, National Instruments. http://www.natinst.com
8. JESS, http://herzberg.ca.sandia.gov
9. C Language Integrated Production System (CLIPS), http://www.jsc.nasa.gov/~clips/CLIPS.html
10. Java Media Framework, Speech API, JavaSpace , Java Activator http://java.sun.com/javaone/javaone98/sessions/
11. Tengah JDBC & Zero Administration Client, WebLogic Inc, http://www.weblogic.com
12. C. Sun, L. Picinich and M. Windrem. *Application of World-Wide-Web in Payload Operations*, Fourth International Symposium on Space Mission Operations and Ground Data Systems (SpaceOps 96), Munich, Germany, 1996. http://www.esoc.esa.de/external/mso/SpaceOps/233/2 33.html